HW4 - EL2700

Mustafa Al-Janabi
970101-5035
musaj@kth.se

Muhammad Zahid
951102-4730
mzmi@kth.se

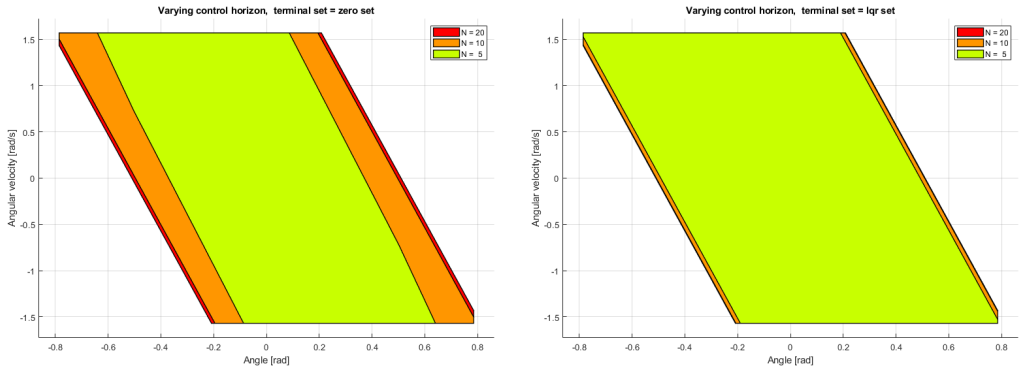
Part I: Sets and constraints

Q1 - Influence of terminal sets

When the size of the terminal set is large, as in the lqr case in Fig. 1 right, that N-step controllable set is larger as opposed to the zero terminal set on Fig. 1 left. This makes sense because if the terminal set is large , the final state can take multiple values which can be reached in fewer steps.

Q2 - Study the influence of the control horizon

If we look at the control invariant set for the LQR terminal set when N changes in Fig.1, we can see that when N increases the size of the control invariant set increases up to a point after which it seems to converge. In the plot, N was changed from 5 to 20 in three steps. This conclusion is true for both the zero and LQR terminal set.

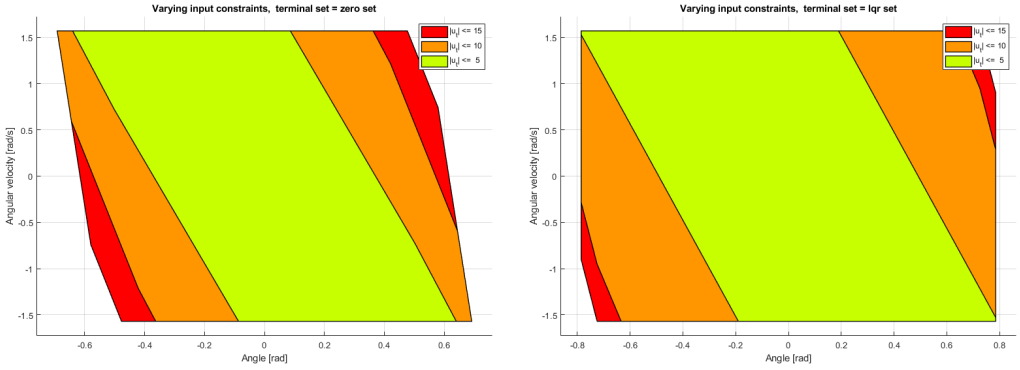


(a) zero set (b) lqr set

Figure 1: Comparing the zero terminal set with the lqr terminal set with respect to the control horizon

Q3 - Influence of control constraints

We then observed the effect of changing the control constraints on the control invariant set while keeping $N = 5$. The results are shown in Fig.2, from which it can be observed that as the control constraints are relaxed from 5 to 15 the control invariant set increases, because the control and then drive more initial states to the desired output.



(a) zero set (b) lqr set

Figure 2: Comparing the zero terminal set with the lqr terminal set with respect to the control input contraints

Part II - MPC for Inverted Pendulum Stabilization

Q4 - Understanding the MPC implementation

1. The state constraints are set in the following code block.

```
# State constraints
if xub is not None:
    con_ineq.append(x_t)
    con_ineq_ub.append(xub)
    con_ineq_lb.append(np.full((self.Nx,), -ca.inf))
if xlb is not None:
    con_ineq.append(x_t)
    con_ineq_ub.append(np.full((self.Nx,), ca.inf))
    con_ineq_lb.append(xlb)
```

Basically in the code above, x_t comes from the optimization variables set by opt_var and we extract the “x” at time t from it. Three lists are updated at each time step; one for the the ”x” at the current step, and other two for the upper and lower bound for this state. These list constraints are then vertically concatenated with the other contrainted vectors and the fed to the solver function as arguments. The solver uses ca.nlpsol. The upper and lower bound of the states are fed to the MPC upon instantiating.

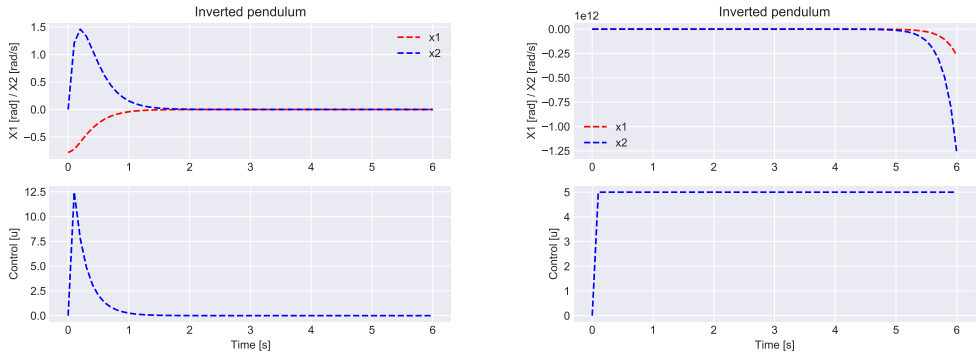
2. The objective function is calculated according to equation 5 in the assignment description. The objective function is updated at each time step to add the current cost to the previous cost. And $x_t - x_{0ref}$ is used because we want to penalize the difference between the current and the desired state. This is done in the for loop over the time horizon on the following line

   ```
   obj += self.running_cost((x_t - x0_ref), self.Q, u_t, self.R)
   ```

   Running cost itself is a Casadi function object. We call it with the state error, the Q weight matrix, the input at time t and the R weight matrix as inputs
3. The terminal cost is added to the total objective at the last time step and it uses another Casadi function object called *terminal_cost*. The procedure is very similar as for the previous point.
4. The *params* holds the initial state x_0, reference x0_ref and input u_0 at the current calculation instance of the MPC solution.
5. We are only taking *u_pred[0]* (only the first calculated input for the optimal solution) because that is basically what MPC is all about. In the next time instance we calculate the problem again with a new initial state and we only take the first calculated input, because we will recalculate the input for the next time step instead of taking it from the current calculation. This works as a feedback for the MPC controller, because it is calculated the input at each time instance, taking into account the current measured states.

Q5 - Comparing input constraints

Using $Q = R = P = 1$, when we set the input limits to ± 5, the system becomes unstable as shown in Fig. 3 with the initial conditions $x_0 = [-\pi/4 \quad 0]$. This is intuitive because from the zero control invariant set, shown in Fig. 1, we can see that the initial condition cannot be stabilized with such a constrained input.



(a) Using $|u_{min}| = |u_{max}| = 20$ (b) Using $|u_{min}| = |u_{max}| = 5$

Figure 3: The MPC solution for different constraints on the input signal.

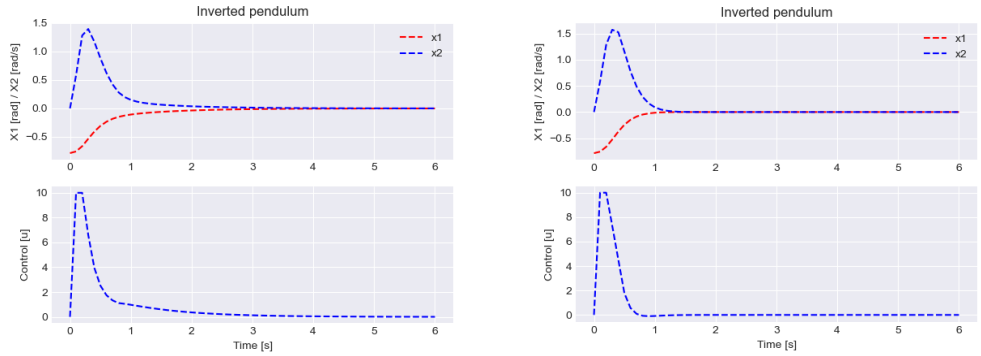
Q6 - lqr vs zero set

The simulation of the system with different terminal sets is shown in Fig. 4. On the left we can see the lqr terminal set which converges to zero slower than the zero terminal set shown in the right. Is is because lqr set does not try to go to zero, but a region around zero. On the other hand, $x1$ state goes higher in the zero set (above 1.5).

Part III - MPC for Pendulum-Cart System

Q7 - MPC reference tracking

We set the cart position reference to $10m$, and set $Q = R = P = 1$. We set the upper and lower limits of $x_3 = \pm \frac{10}{180}\pi$ to full-fill the requirement of having the angle at $\pm 10°$ from the upright position.



(a) Using lqr terminal set

(b) Using zero terminal set

Figure 4: The MPC solution for different terminal set with $|u_{min}| = |u_{max}| = 10$ and 5 steps.

From the results in Fig. 5 we can see that all the requirements are met and the pendulum with cart stabilizes within $6s$.

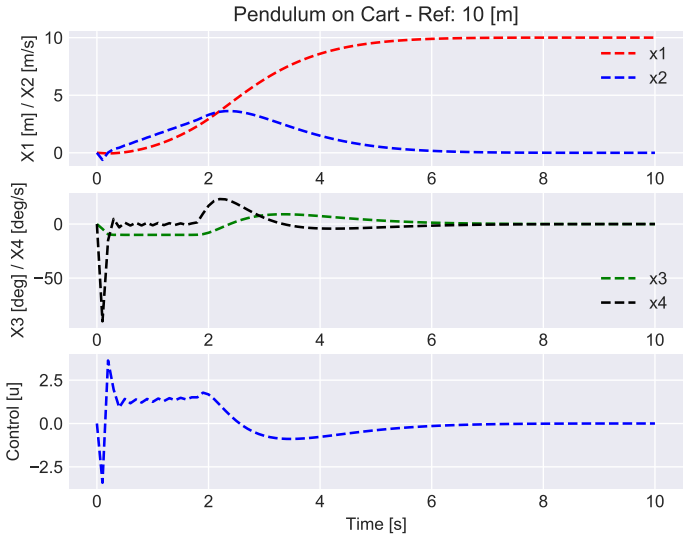
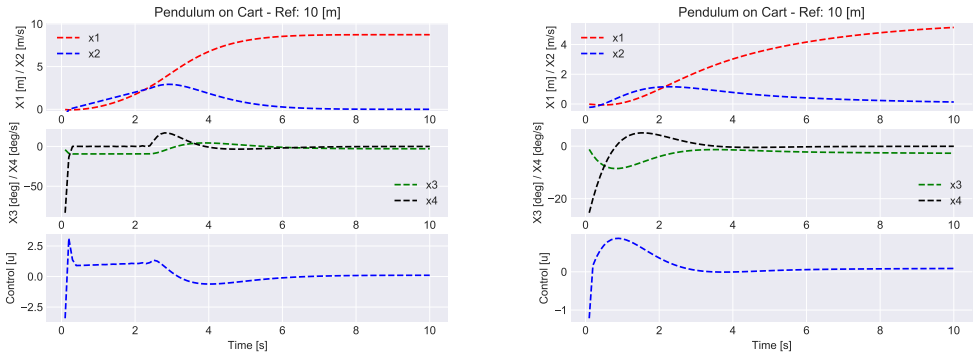


Figure 5: Reference tracking performance of the MPC controller

Q8 - MPC vs LQR with disturbance

Next, we set the disturbance in Q2 Assignment 3 to $w = 0.1$. The results is shown in Fig. 6, with the MPC controller on the left and the LQR controller on the right. Comparing the plots, we can see that the MPC controller tries to eliminate the disturbance better than the LQR controller, which is shown by a lower steady-state error. In the MPC controller we can still see a small steady state error that can be fixed by augmenting the system matrix to include integral action like we did in assignment 3.



(a) MPC

(b) LQR

Figure 6: Reference of the MPC and the LQR controllers with disturbance $w = 0.1$